

What is claimed is:

1. A method comprising:

determining, during a process of booting a processing system, whether
5 sufficient resources are available to meet resource requests for multiple devices
associated with a peripheral connect interface (PCI) subsystem of the processing
system;

retrieving boot information from an extended firmware interface (EFI)
environment of the processing system;

10 automatically identifying at least one of the multiple devices as boot-critical,
based on the boot information from the EFI environment; and

if sufficient resources to meet the resource requests are not available,
automatically allocating resources for the device identified as boot-critical but not for
all of the devices associated with the PCI subsystem, before the processing system
15 boots an operating system (OS).

2. A method according to claim 1, further comprising:

identifying at least one device associated with the PCI subsystem as not
critical to booting the processing system; and

20 disabling the device identified as not critical, before the processing system
boots the OS, if sufficient resources to meet the resource requests are not available.

3. A method according to claim 1, wherein the operation of retrieving boot
information from an EFI environment of the processing system comprises:

25 retrieving information that designates at least one device selected from a
group consisting of:

a device to serve as an input console during boot;

a device to serve as an output console during boot; and

a device to provide OS boot instructions.

30

4. A method according to claim 1, further comprising:
discovering the devices associated with the PCI subsystem by scanning the
PCI subsystem.

5 5. A method according to claim 1, further comprising:
collecting resource requests from at least two of the devices associated with
the PCI subsystem; and
determining whether sufficient resources are available, based on the collected
resource requests.

10

6. A method according to claim 1, further comprising:
determining available resources of the processing system, based on resource
information from a host bridge driver.

15 7. A method according to claim 6, wherein the operation of determining whether
sufficient resources are available to meet the resource requests comprises:
automatically determining whether sufficient resources are available, based
on the resource information from the host bridge driver.

20 8. A method according to claim 1, further comprising:
automatically identifying at least one of the devices associated with the PCI
subsystem to be rejected, based on a rejection policy.

9. A method according to claim 1, further comprising:

collecting resource requests from at least two of the devices associated with the PCI subsystem;

5 determining available resources of the processing system, based on resource information from a host bridge driver.

automatically determining whether sufficient resources are available for the devices associated with the PCI subsystem, based on the collected resource requests and the resource information from the host bridge driver; and

10 disabling at least one of the devices associated with the PCI subsystem before the processing system boots the OS, if sufficient resources are not available to meet the resource requests.

10. An apparatus comprising:

a processor;

15 a peripheral connect interface (PCI) subsystem communicatively coupled to the processor;

two or more devices associated with the PCI subsystem;

a machine accessible medium communicatively coupled to the processor; and

20 a bus driver encoded in the machine accessible medium, wherein the bus driver, when executed by the processor, performs operations comprising:

determining, during a process of booting the processing system, whether sufficient resources are available to meet resource requests for the two or more devices;

25 retrieving boot information from an extended firmware interface (EFI) environment of the processing system;

automatically identifying at least one of the devices as boot-critical, based on the boot information from the EFI environment; and

30 if sufficient resources to meet the resource requests are not available, facilitating successful booting of an operating system (OS) by automatically allocating resources for the device identified as boot-critical but not for all of the devices associated with the PCI subsystem, before the processing system boots the OS.

11. An apparatus according to claim 10, wherein the bus driver performs further operations, comprising:

identifying at least one device associated with the PCI subsystem as not critical to booting the processing system; and

5 disabling the device identified as not critical, before the processing system boots the OS, if sufficient resources to meet the resource requests are not available.

12. An apparatus according to claim 10, wherein the bus driver retrieves boot information from the EFI environment designating at least one device selected from a group consisting of:

a device to serve as an input console during boot;

a device to serve as an output console during boot; and

a device to provide OS boot instructions.

13. An apparatus according to claim 10, wherein the bus driver performs further operations, comprising:

discovering the devices associated with the PCI subsystem by scanning the PCI subsystem.

14. An apparatus according to claim 10, wherein the bus driver performs further operations, comprising:

collecting resource requests from at least two of the devices associated with the PCI subsystem; and

25 determining whether sufficient resources are available, based on the collected resource requests.

15. An apparatus according to claim 10, wherein:

the apparatus further comprises a host bridge driver encoded in the machine accessible medium; and

30 the bus driver determines available resources of the processing system, based on resource information obtained from the host bridge driver.

16. An apparatus according to claim 15, wherein the bus driver automatically

determines whether sufficient resources are available, based on the resource information obtained from the host bridge driver.

17. An apparatus according to claim 10, wherein, if sufficient resources are not available, the bus driver automatically identifies at least one of the devices associated with the PCI subsystem to be rejected, based on a rejection policy.

18. An apparatus according to claim 10, wherein:
the apparatus further comprises a host bridge driver encoded in the machine accessible medium; and
the operations performed by the bus driver further comprise:
collecting resource requests from at least two of the devices associated with the PCI subsystem;
determining available resources of the processing system, based on resource information from the host bridge driver;
automatically determining, based on the collected resource requests and the resource information from the host bridge driver, whether sufficient resources are available for the devices associated with the PCI subsystem; and
disabling at least one of the devices associated with the PCI subsystem, before the processing system boots the OS, if sufficient resources are not available to meet the resource requests.

19. An apparatus comprising:
a machine accessible medium; and
instructions encoded in the machine accessible medium, wherein the instructions, when executed by a processing system, cause the processing system to perform operations comprising:
determining, during a process of booting the processing system, whether sufficient resources are available to meet resource requests for two or more devices associated with a peripheral connect interface (PCI) subsystem of the processing system;
retrieving boot information from an extended firmware interface (EFI) environment of the processing system;

automatically identifying at least one of the devices as boot-critical, based on the boot information from the EFI environment; and

if sufficient resources are not available to meet the resource requests, automatically allocating resources for the boot-critical device but not for all of the devices associated with the PCI subsystem, before the processing system boots an operating system (OS).

20. An apparatus according to claim 19, further comprising:

a bus driver encoded in the machine accessible medium, wherein the bus driver comprises the instructions.

21. An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

identifying at least one device associated with the PCI subsystem as not critical to booting the processing system; and

disabling the device identified as not critical, before the processing system boots the OS, if sufficient resources to meet the resource requests are not available.

22. An apparatus according to claim 19, wherein the operation of retrieving boot information from an EFI environment of the processing system comprises:

retrieving information that designates at least one device selected from a group consisting of:

a device to serve as an input console during boot;

a device to serve as an output console during boot; and

a device to provide OS boot instructions.

23. An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

discovering the devices associated with the PCI subsystem by scanning the PCI subsystem.

24. An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

collecting resource requests from at least two of the devices associated with the PCI subsystem; and

determining whether sufficient resources are available, based on the collected resource requests.

5

25. An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

determining available resources of the processing system, based on resource information from a host bridge driver.

10

26. An apparatus according to claim 25, wherein the operation of determining whether sufficient resources are available comprises:

automatically determining whether sufficient resources are available, based on the resource information from the host bridge driver.

15

27. An apparatus according to claim 19, wherein:

if sufficient resources are not available, the instructions automatically identify at least one of the devices associated with the PCI subsystem to be rejected, based on a rejection policy.

20

28. An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

collecting resource requests from at least two of the devices associated with the PCI subsystem;

25

determining available resources of the processing system, based on resource information from a host bridge driver;

automatically determining, based on the collected resource requests and the resource information from the host bridge driver, whether sufficient resources are available for the devices associated with the PCI subsystem; and

30

disabling at least one of the devices associated with the PCI subsystem, before the processing system boots the OS, if sufficient resources are not available to meet the resource requests.